

## **AMENDMENTS TO THE CLAIMS**

This listing of claims replaces all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) A computer-implemented diagram system for manipulating a domain specific language instance model via a diagram, comprising:
  - a diagram that stores at least one shape element;
  - an object model application program interface that includes a control that facilitates access to the diagram, the control maintaining state information associated with the diagram and with an underlying data document, employing a common framework to keep both the diagram and the underlying data document synchronized; and
  - a computer readable storage medium comprising sets of code and data structures for causing a computer to modify the diagram,wherein the underlying data document is a domain specific language model representing both model elements and presentation elements.
2. (Previously Presented) The computer-implemented system of claim 1, the state information comprising at least one of selection, zoom and scroll position.
3. (Previously Presented) The computer-implemented system of claim 1, the control captures operating system events.
4. (Previously Presented) The computer-implemented system of claim 3, the control providing at least some of the operating system events to the shape element.
5. (Previously Presented) The computer-implemented system of claim 1, the control rerouting at least one of paint, keyboard and mouse events to at least one of the diagram and the shape element.
6. (Previously Presented) The computer-implemented system of claim 1, the

diagram and the shape element responsible for painting themselves.

7. (Previously Presented) The computer-implemented system of claim 1, the diagram or the shape element being responsible for responding to a user interaction.
8. (Previously Presented) The computer-implemented system of claim 1, the shape element being based, at least in part, upon a model element class.
9. (Previously Presented) The computer-implemented system of claim 1, the diagram being based, at least in part, upon a node shape that has a bounds property which defines its location and size, the node shape derived from the shape element.
10. (Previously Presented) The computer-implemented system of claim 1, at least one shape element having a child shape element.
11. (Previously Presented) The computer-implemented system of claim 1, the shape element derived from a presentation element.
12. (Previously Presented) The computer-implemented system of claim 1, the shape element comprising at least one of a geometry property, a style set property and a shape fields property.
13. (Previously Presented) The computer-implemented system of claim 1, the diagram having a graph object employed for hittesting for testing a shape that has been user dropped by dragging.
14. (Previously presented) The computer-implemented system of claim 1, the shape element being control-less.

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Currently Amended) A computer readable medium storing computer executable components of a diagram system comprising:

a diagram component that stores at least one shape element and has a graph object employed for hittesting for testing a shape that has been user dropped by dragging, wherein the hittesting returns information which indicates if the shape has been dropped outside a boundary of a target shape, on the boundary of the target shape, inside the filed-area of a two dimensional shape and/or on a line of a one-dimensional shape; and,

an application program interface component that includes a control that facilitates access to the diagram component, the control maintaining state information associated with the diagram and with an underlying data document, wherein the application program interface employs a common framework to keep the diagram component and the underlying data document synchronized, wherein the underlying data document is a domain specific language model representing both model elements and presentation elements.

20. (Cancelled)

21. (Currently Amended) A method for diagramming, comprising:

managing presentation elements comprised of diagrams and shapes in a same context as correspondingly depicted design elements of a diagram on design surface in an object model diagramming system to avoid synchronization issues of mirrored presentation and design classes;

providing an object model application programming interface comprising a single diagram control for the design surface that maintains state information associated with the diagram by capturing events; and

rendering shapes of the diagram that are responsible for painting themselves and for responding to user interaction via a user interface, making implementation very light weight and independent of any specific diagram, wherein at least a one shape is a parent shape having an associated child shape, wherein the child shape inherits resources from the parent shape, the child shape conforming to a set of rules defining where the child shape can be located and how the child shape can be sized relative to the parent shape, wherein the parent shape and the child shape may be in a nested relationship such that the child shape is fully contained within the parent shape.

22. (New) A computer program product for use at a computer system, the computer program product for implementing a method for a diagram system to manipulate a domain specific language instance model via a diagram, the computer program product comprising one or more physical computer-readable storage media having stored thereon computer-executable instructions that when executed by a processor cause the computer system to perform the following:

providing a diagram that stores at least one shape element;

providing an object model application program interface that includes a control that facilitates access to the diagram, the control maintaining state information associated with the diagram and with an underlying data document, employing a common framework to keep both the diagram and the underlying data document synchronized; and

providing a computer readable storage medium comprising sets of code and data structures for causing a computer to modify the diagram,

wherein the underlying data document is a domain specific language model representing both model elements and presentation elements.

23. (New) A computing system comprising the following:

one or more processors;

system memory;

one or more computer-readable storage media having stored thereon computer-executable instructions that, when executed by the one or more processors, causes the computing system to perform a method for a diagram system to manipulate a domain specific language instance model via a diagram, the method comprising the following:

providing a diagram that stores at least one shape element;

providing an object model application program interface that includes a control that facilitates access to the diagram, the control maintaining state information associated with the diagram and with an underlying data document, employing a common framework to keep both the diagram and the underlying data document synchronized; and

providing a computer readable storage medium comprising sets of code and data structures for causing a computer to modify the diagram,

wherein the underlying data document is a domain specific language model representing both model elements and presentation elements.